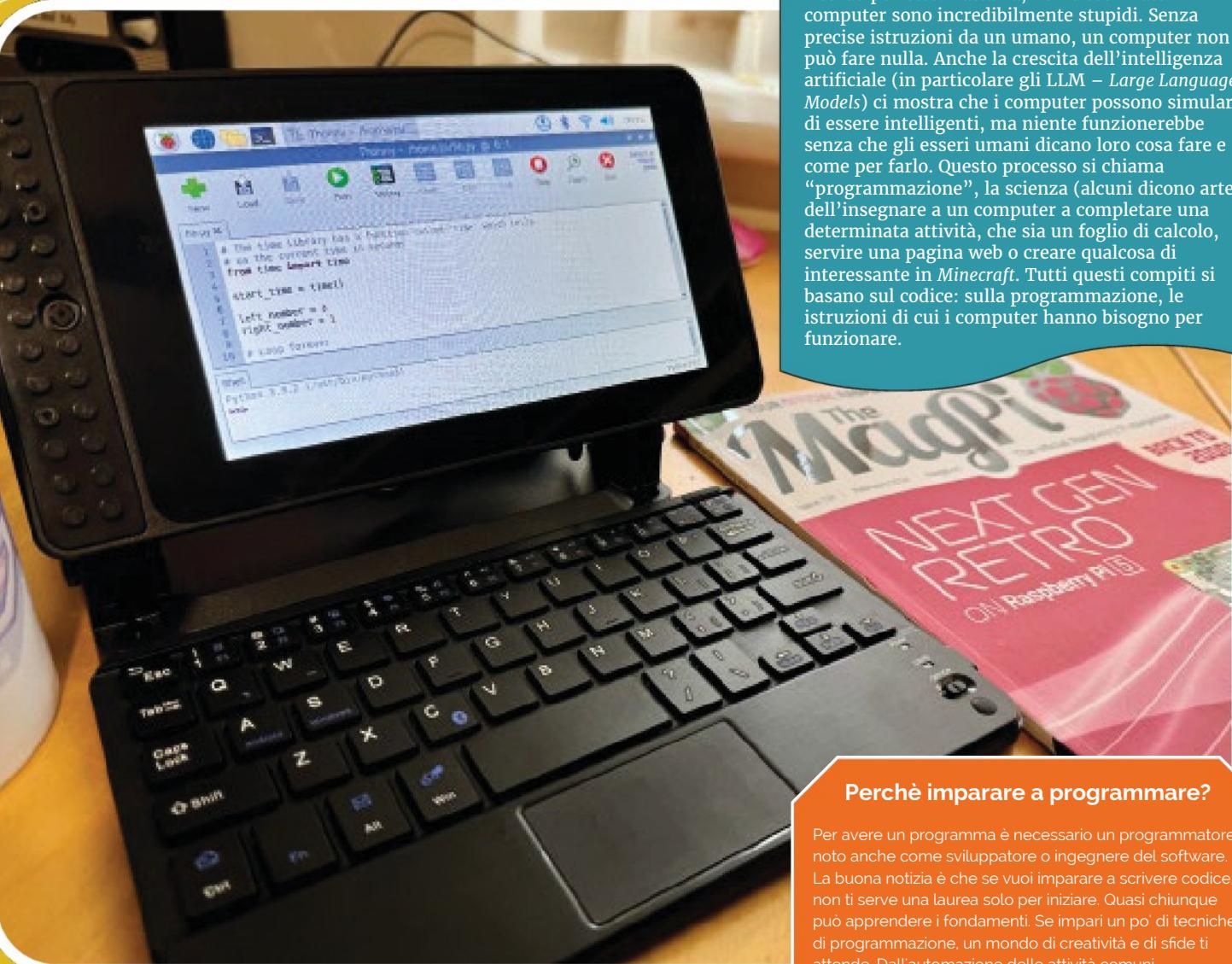
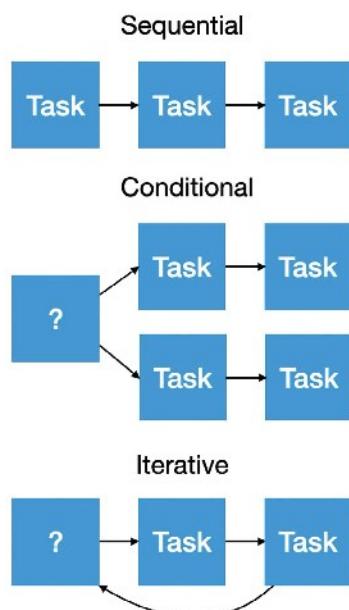
The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Mount Pleasant House, Cambridge, CB3 0RN. ISSN: 2051-9982.

PROGRAMMA con RASPBERRY PI

La programmazione è una delle attività più gratificanti cose che puoi fare con Raspberry Pi.
Ma da dove cominciare?
PJ Evans prende la tastiera

> "Tutti i computer sono incredibilmente stupidi. Senza precise istruzioni da un umano, un computer non può fare nulla"

Spesso si pensa ai computer come cose estremamente intelligenti, forse molto più dell'essere umano medio. Dopotutto, possono pensare più velocemente di noi e dei computer sono inarrivabili nella matematica. Possono ricordare più di noi e hanno sempre un ricordo perfetto. Tuttavia, non è così. Tutti i computer sono incredibilmente stupidi. Senza precise istruzioni da un umano, un computer non può fare nulla. Anche la crescita dell'intelligenza artificiale (in particolare gli LLM – *Large Language Models*) ci mostra che i computer possono simulare di essere intelligenti, ma niente funzionerebbe senza che gli esseri umani dicano loro cosa fare e come per farlo. Questo processo si chiama "programmazione", la scienza (alcuni dicono arte) dell'insegnare a un computer a completare una determinata attività, che sia un foglio di calcolo, servire una pagina web o creare qualcosa di interessante in *Minecraft*. Tutti questi compiti si basano sul codice: sulla programmazione, le istruzioni di cui i computer hanno bisogno per funzionare.

Perchè imparare a programmare?

Per avere un programma è necessario un programmatore, noto anche come sviluppatore o ingegnere del software. La buona notizia è che se vuoi imparare a scrivere codice, non ti serve una laurea solo per iniziare. Quasi chiunque può apprendere i fondamenti. Se impari un po' di tecniche di programmazione, un mondo di creatività e di sfide ti attende. Dall'automazione delle attività comuni all'interazione con l'ambiente fino a creare siti Web o persino giochi, la programmazione è un'abilità essenziale per ottenere il massimo dal tuo computer. Soprattutto (quasi sempre) è molto divertente ed estremamente gratificante quando esegui il tuo programma e funziona.

Scegli il tuo linguaggio

Con alcune eccezioni che tratteremo più avanti, quasi tutta la programmazione si svolge utilizzando linguaggi basati su testo. Esistono moltissimi linguaggi diversi in circolazione e qui tratteremo alcuni dei più popolari. Perché non c'è un solo linguaggio? Per due motivi: in primo luogo, alcune linguaggi sono progettati per compiti specifici, come R per l'analisi dei dati, mentre altri sono noti come linguaggi per uso generale. In secondo luogo, chi progetta questi linguaggi raramente è d'accordo su qualcosa e cerca sempre di migliorare le cose, creandone di nuove.

I linguaggi sono in genere basate sul testo in quanto questo è il modo più semplice per inserire informazioni in un computer. Tutti i linguaggi hanno tre strutture di base: sequenziale, condizionale e iterativo. Esaminiamoli scrivendo un programma (codice che esegue un compito specifico) che prepari una tazza di tè.

▲ I tre tipi fondamentali di struttura del codice

Istruzioni sequenziali

Le istruzioni sequenziali vengono eseguite in sequenza, una dopo l'altra. Ecco alcune istruzioni per preparare una tazza di tè:

1. Riempi il bollitore
2. Accendi il bollitore
3. Prendi tazze, bustine di tè e latte
4. Metti la bustina di tè in tazza con acqua
5. Aggiungi latte

Tuttavia, un computer avrebbe delle difficoltà con queste istruzioni. In primo luogo riempirebbe il bollitore sia che fosse già pieno o meno. Quindi aggiungiamo un condizionale; del codice che prende una decisione:

- Se il bollitore non è pieno di acqua, riempi il bollitore fino all'orlo

Ora non abbiamo l'acqua su tutto il pavimento della cucina. Un altro problema è che il computer non aspetta che il bollitore raggiunga l'ebollizione, quindi avremmo un tè freddo! Iteriamo o eseguiamo continuamente del codice fino a che la condizione è soddisfatta:

- Fino a che il bollitore non bolle, attendi un po' e poi controlla di nuovo

Infine, se volessimo molte tazze di tè, potremmo creare una funzione, o gruppo di comandi, chiamati "Prepara il tè" e poi ripeterlo più volte. Le funzioni sono particolarmente utili se è necessario ripetere di nuovo un'attività. Hai bisogno di organizzare una grande festa? Per 5.000 volte: esegui "Prepara il tè".

Dai un'occhiata a Scratch

Scratch del MIT introduce i concetti di programmazione nel modo più amichevole possibile. Invece del testo, puoi trascinare e rilasciare blocchi colorati per creare programmi semplici e giochi divertenti. È il primo linguaggio di programmazione perfetto e ha fatto parte di Raspberry Pi OS fin dall'inizio.

scratch.mit.edu.

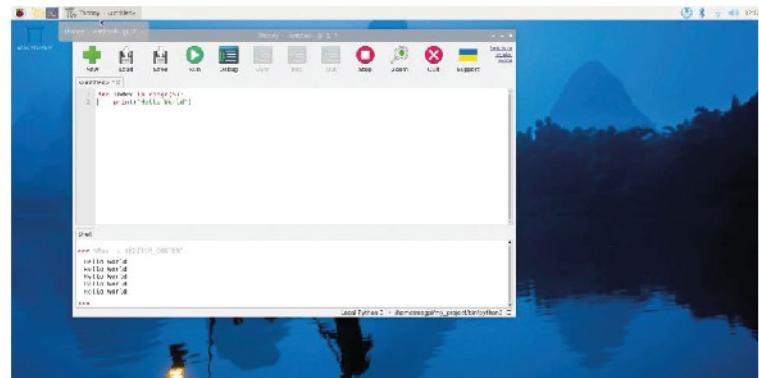
Python

Sapevi che "Pi" in Raspberry Pi è un cenno al linguaggio di programmazione Python? Python è nato come un modo per rendere il codice simile al linguaggio umano ed eliminare tutta la strana punteggiatura di altri linguaggi più diffusi come il C++.

Di conseguenza, Python è senza dubbio il linguaggio più popolare per principianti. Ha una sintassi (il termine per le regole di un linguaggio) amichevole e può essere esteso con moduli che rendono le cose come comunicare con il GPIO di Raspberry Pi il più semplice possibile. Questi moduli hanno portato Python a diventare estremamente influente nell'analisi dei dati e nell'intelligenza artificiale.

Python è un paese delle meraviglie creativo. Interagisci con il mondo reale utilizzando i suoi strumenti GPIO, crea giochi con la libreria Pygame, crea applicazioni Web e altro. Non è una coincidenza che la maggior parte degli articoli di programmazione in questa rivista siano basati su Python. Python è di serie con tutti i sistemi operativi Raspberry Pi, insieme a Thonny, una applicazione (nota come IDE) che ti aiuta a interagire con il linguaggio.

Creiamo il nostro primo programma. Apri Thonny (Menu Raspberry Pi> Programmazione> Thonny IDE e inserisci le due righe di codice di **Hello_World. Py**. Nota che quando premi **INVIO** dopo la prima riga, Thonny rientrerà in automatico la seconda riga (spiegheremo tutto questo in un tutorial più avanti nella rivista). Per ora sappi che il rientro è di quattro spazi. Controlla che ogni singolo carattere sia scritto esattamente come lo vedi nel listato.

Fai clic su Esegui e vedrai l'output "Ciao Mondo!"nella shell sottostante cinque volte. Congratulazioni! Hai scritto un programma per computer.

python™

> “Python è senza dubbio il linguaggio più popolare per principianti”

hello.py

▸ Linguaggio: **Python**

```
001.  for index in range(5):
002.      print("Ciao Mondo!")
```

Node.js

JavaScript è uno dei linguaggi di programmazione più popolari al mondo. Ha iniziato la sua vita sui browser web ed è il linguaggio usato per creare siti Web interattivi. Se visiti un sito Web, è probabile che il tuo browser esegua JavaScript per farlo funzionare. Node.js è un sistema per l'esecuzione di javascript a livello locale, piuttosto che nel browser web. Le app Node alimentano il back-end dei servizi Web di tutto il mondo.

È meno amichevole di Python con una sintassi simile a C++, ma è anche molto più veloce. Node.js attira gli sviluppatori poiché bilancia bene velocità e complessità. Come Python, Node.js dispone di moduli che possono aggiungere tutti i tipi di funzionalità al linguaggio.

Quindi, se sei interessato a scrivere API (Application Programming Interfaces – utilizzate da browser Web per inviare e ricevere i dati direttamente dai server Internet) o siti Web, node.js è dove devi cercare. Ora è ben supportato da Raspberry Pi ed è incredibilmente veloce sul nuovo Raspberry Pi 5.

hello.js

Linguaggio: **Node.js**

```
001.  for(let index = 0; index < 5; index++) {
002.     console.log('Ciao Mondo!');
003.  }
```

Go

Un nuovo attore sul campo, GO è stato sviluppato dagli ingegneri di Google stufi delle incoerenze tra i diversi linguaggi (i programmatori non sono mai d'accordo su queste cose). Python e Javascript sono interpretati in fase di esecuzione, il che significa che il codice è compilato durante l'esecuzione. Questo ha molti vantaggi, soprattutto per gli sviluppatori, ma causa un significativo sovraccarico e rallenta le operazioni.

GO può essere sia interpretato che compilato in fase di esecuzione, il che significa che il suo codice può essere trasformato in codice a livello di codice macchina, che offre un enorme aumento della velocità.

Rispetto a un linguaggio più complesso, Go è più facile da usare e più facile da imparare rispetto alla maggior parte dei suoi pari. Può essere usato praticamente per qualsiasi scopo ed è ben supportato su Raspberry Pi. Google si è concentrata su una chiara sintassi che combina la brevità di Javascript con la leggibilità di Python. Go ora sta iniziando a intaccare il dominio di Node nel mondo dei server web, quindi è un linguaggio che merita la tua attenzione.

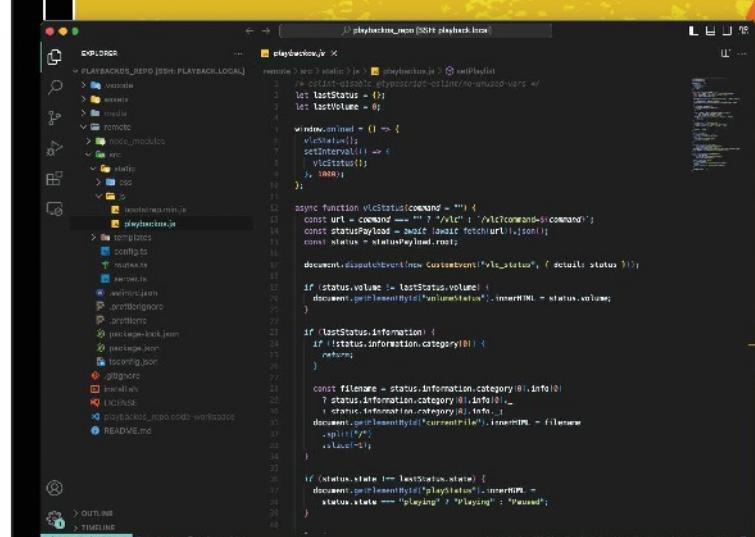
hello.go

Linguaggio: **Go**

```
001.  package main
002.  import "fmt"
003.  func main() {
004.     for index := 0; index < 5; index++ {
005.        fmt.Println("Ciao Mondo!")
006.     }
007.  }
```

Strumenti di programmazione e suggerimenti

Puoi scrivere il codice con qualsiasi editor di testo o anche direttamente nei documenti usando il terminale. Tuttavia, probabilmente vorrai utilizzare un editor di testo dedicato o IDE (Ambiente di Sviluppo Integrato).

Gli IDE sono app tipo elaboratori di testo che aiutano la scrittura del codice rilevando bug e errori di sintassi e aiutano nelle strutture di comando. Hanno spesso plug-in per aiutarti specificamente con il linguaggio di tua scelta.

Installare Visual Studio Code

Senza dubbio, il più popolare IDE del mondo è Visual Studio Code (noto anche come "VS Code"). Sebbene non sia strettamente un vero IDE, è un editor di testo con così tante funzionalità da sfumare il confine tra i due concetti. Una volta che ti sei diplomato in Scratch e Thonny, VS Code è la tua prossima destinazione.

Oltre alla codifica a colori della sintassi per aiutarti a leggere il tuo codice, VS Code viene fornito con una impressionante gamma di scorciatoie da tastiera che rendono la vita di uno sviluppatore molto più semplice. Ma dove brilla davvero, è il Marketplace delle estensioni, che ti consente di personalizzare la tua esperienza in misura sorprendente. Poiché chiunque può inviare un'estensione al marketplace, se vuoi che VS Code faccia qualcosa, è probabile che esista già. Dalla codifica a colori dei file CSV alla distribuzione di servizi Web o alla formattazione automatica del codice per mantenerlo in ordine, VS Code è un coltellino svizzero per qualsiasi sviluppatore.

Abbiamo già detto che è gratuito? Lo stesso vale per la maggior parte delle estensioni. Per iniziare è possibile installare VS Code su Raspberry Pi OS (desktop) dall'app Applicazioni Raccomandate. Una delle nostre estensioni preferite è SSH, che ti consente di sviluppare ed eseguire il codice su un Raspberry Pi da qualsiasi altra macchina sulla rete, così puoi programmare il tuo progetto su qualsiasi computer come se utilizzassi il Raspberry Pi stesso.

“ Senza dubbio, il più popolare IDE del mondo è Visual Studio Code ”

Imparare a debuggare

I bug succedono. È una delle prime lezioni che impari. Gli sviluppatori professionisti vedono i bug come parte del processo; scoprire cosa non funziona. Quindi non scoraggiarti quando il codice non viene eseguito la prima volta.

Un bug può assumere molte forme, che si tratti di semplice battitura che impedisce l'esecuzione del codice o di un errore di calcolo che significa che il codice viene eseguito, ma produce il risultato sbagliato.

La rimozione di bug dal codice (debug) è una parte fondamentale dell'esperienza di qualsiasi programmatore. La lotta si divide in due parti: trovare il pezzo di codice che non si comporta correttamente e poi risolverlo. Ognuna ha le sue sfide.

Se il tuo codice non si comporta come previsto, vedi se l'editor o l'IDE che stai usando ha un debugger, la capacità di eseguire il codice riga per riga o fermarlo ad un certo punto in modo da poterlo ispezionare. Questo passo però può portare a un momento "ah ha!" in cui il codice si dirige in una direzione diversa dal previsto.

Una tecnica collaudata consiste nel monitorare l'output del codice aggiungendo istruzioni di registro. Ad esempio, in un'app da linea di comando Python, l'aggiunta di **print ("QUI")** o **print (variabile)** può aiutarti a tenere traccia di ciò che sta succedendo. È una tecnica rudimentale, ma milioni di programmatori la usano ogni giorno per trovare problemi.

Quando hai trovato il bug, la sua soluzione potrebbe non essere ovvia. A volte può essere, ad esempio "Oh, ho usato un + anziché un - per calcolare il totale". Se ti ritrovi bloccato, potrebbe essere necessario eseguire il refactoring: il processo di scarto di una sezione di codice e riscrittura per fornire una soluzione migliore.

▲ Esempio di output di errore dalla console di debug di VS Code

Il gergo

Alcuni termini comuni che incontrerai

Variabile
Una etichetta che rappresenta un testo o un valore numerico. Utilizzato per archiviare informazioni per un uso successivo.

```
fetteDiPizzaAiPeperoni = 10
```

Funzione
Parte di codice raggruppato per eseguire una funzione che può essere usato più volte senza che lo tu lo debba ripetere

```
def faiPizza(tipo):
        Ingredienti = pizze[tipo].condimenti;
```

Condizionali, If
Un pezzo di codice che prende una decisione

```
If (tipo.formaggio):
        aggoingiFormaggio()
```

Cicli
Un'istruzione che ripete un blocco di codice fino a che alcune condizioni vengono soddisfatte

```
for (condimento in condimenti):
        aggiungiIngrediente(condimento)
```

Classi e Oggetti
Una raccolta di variabili e funzioni strutturate in un gruppo logico che combina sia variabili che funzioni.

Array, Dizionari, Liste
Un elenco di variabili correlate

```
pizze = [margherita,  napoli, zola]
```

Moduli, Estensioni
La possibilità di aggiungere funzionalità nuove o migliorate ad un linguaggio di programmazione utilizzando il codice pubblico esistente

```
from pizzalib import faiPizza
```

Compilazione
Convertire il testo del codice in codice binario eseguibile per un'esecuzione più rapida

Suggerimenti di partenza

- **Senza paura**
Non importa se qualcosa non va più, semplicemente ricomincia. Non puoi danneggiare il computer.

- **Parti in piccolo**
Non provare a scrivere un concorrente di Excel la prima volta. Trova un piccolo problema e vedi se riesci a risolverlo.

- **Tienilo semplice**
Non scrivere mai più codice di quel che devi. In programmazione questo è noto come "over-engineering" o YAGNI (non ti servirà).

- **Preoccupazioni separate**
Ciò significa che qualsiasi blocco di codice dovrebbe fare un lavoro e uno solo. Fare altrimenti è creare "Spaghetti coding".

- **Cercalo**
Alcuni sviluppatori professionisti scherzando dicono che la loro vera abilità è usare i motori di ricerca. C'è un mare di informazioni e se sei rimasto bloccato, è probabile che qualcun altro abbia già risolto lo stesso problema.

- **Non ripeterti (DRY)**
Se scopri che stai ripetendo lo stesso codice più volte, inseriscilo in una funzione e richiamala. Farlo significa avere un posto solo dove correggere i bug.

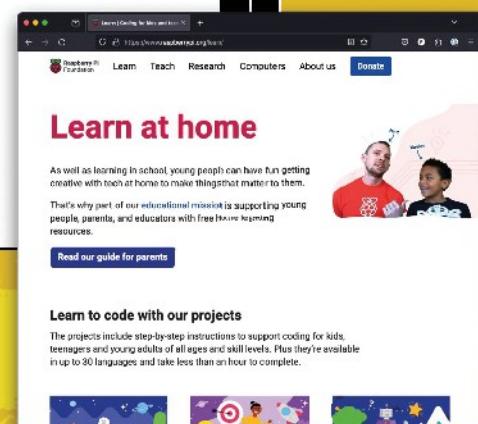
- **Dai nomi sensati**
Un altro principio comune è "dare un nome alle cose è difficile". In verità non lo è, ma alla gente piacciono le scorciatoie. Non chiamare mai una variabile "x" o "y", racconta invece una storia: "GranTotaleOrdinePizza"ti dice tutto ciò che devi sapere.

Risorse per l'insegnamento

Ci sono molti "campi di addestramento" online per imparare il codice e coprono tutti i livelli di esperienza. Qui ci sono alcuni dei nostri preferiti:

- **FreeCodeCamp**
Una enorme, e gratuita come suggerisce il nome, collezione di corsi adatti per i principianti assoluti
freecodecamp.org

- **Raspberry Pi resources**
La nostra Fondazione Raspberry Pi è una risorsa, soprattutto per scratch e Python
raspberrypi.org/learn

- **Learn Python**
Un corso completo di Python con il quale programmi nel browser!
learnpython.org

- **Nodeschool**
Impara Node gratuitamente con questa raccolta di Workshop autonomi
nodeschool.io

- **Go Learn**
La risorsa ufficiale del progetto Go con disponibili dei tutorial per principianti
go.dev/learn

Codice quotidiano

Come nell'apprendimento di qualsiasi nuova abilità, la pratica quotidiana produce benefici e aumenta la tua "memoria muscolare" per la programmazione. Prova a scrivere una piccola app ogni giorno. Se hai bisogno di ispirazione, ci sono molte sfide di programmazione in giro come il Free Code Camp **freecodecamp.org** o il più complicato Advent Of Code: **adventofcode.com**

È importante non farsi intimidire dalla potenziale complessità dell'imparare un linguaggio di programmazione. Poni un piccolo obiettivo e costruisci da lì. Non è un caso che la prima cosa che si fa è scrivere "Ciao Mondo!" sullo schermo. Hai appena scritto il tuo primo programma! Programmare può e deve essere divertente e gratificante.

Parte 01

Imparare Python: Fare una To-Do List

Tieni traccia delle tue attività con un semplice elenco delle cose da fare. Questo programma spiega le nozioni di base della programmazione Python

MAKER
Lucy Hattersley

Lucy è capo redattrice di *The MagPi* e adora una buona To Do List.
magpi.cc

Cosa Serve

- Raspberry Pi
- Raspberry Pi OS
- Thonny IDE

Python è un linguaggio di programmazione incredibilmente versatile. È fantastico per i principianti perché è facile da leggere e capire, ma è anche abbastanza potente per i data scientist ed esperti di IA. Anche se sono disponibili molti altri linguaggi di programmazione e ognuno ha i suoi pregi a seconda di quel che stai cercando di fare, pensiamo che Python sia quello da cui iniziare.

In questo tutorial, ti mostreremo come creare un'applicazione per l'elenco delle cose da fare in Python. Non sarà la migliore lista di cose da fare al mondo. Non è questo il punto. Lo farai da solo e inizierai a comprendere i mattoni fondamentali dei linguaggi di programmazione.

Ecco alcuni dei concetti che esamineremo

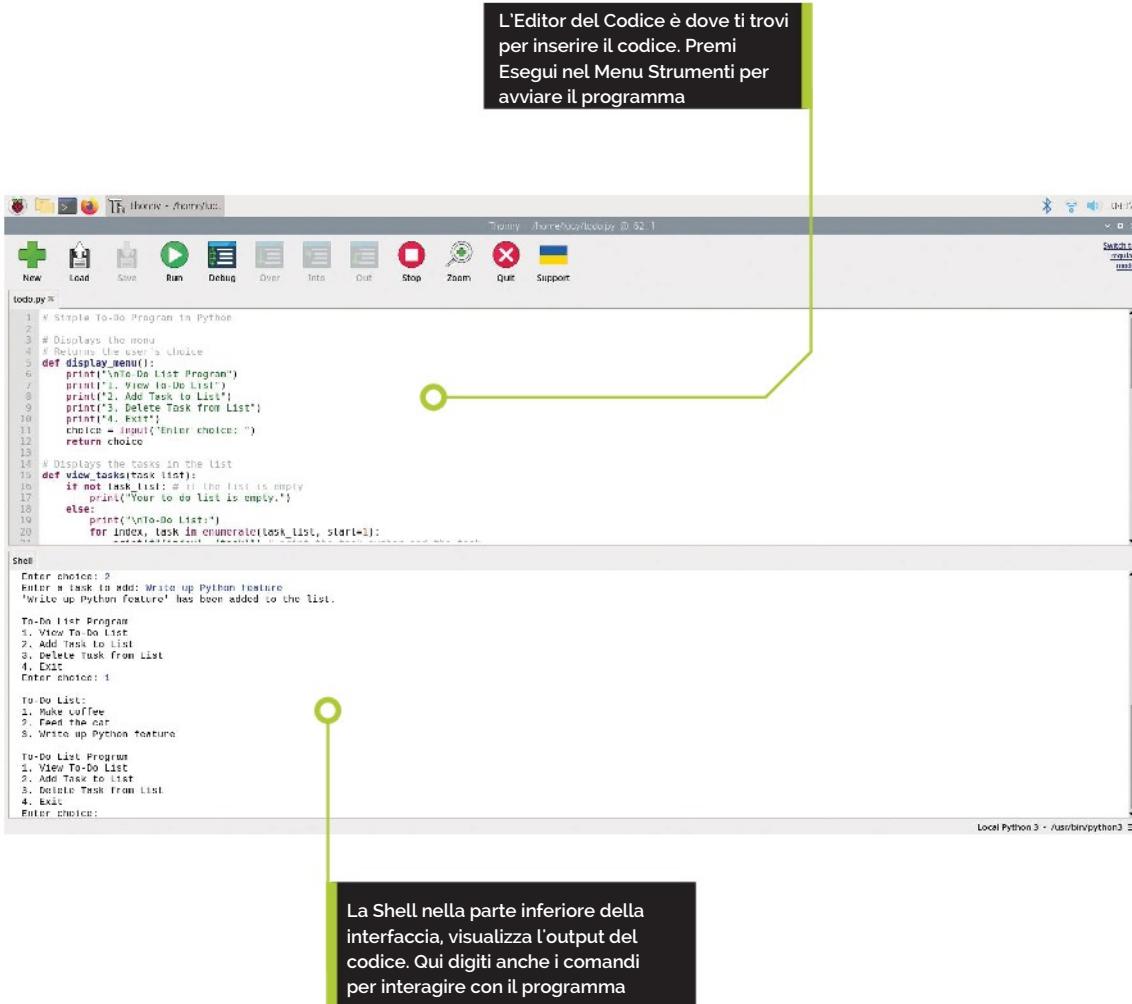
- **Variabili:** vengono utilizzate per memorizzare degli elementi (in questo caso le nostre cose da fare)
- **Funzioni:** servono per ripetere velocemente il codice
- **Condizioni:** Utilizzate per indirizzare il codice in direzioni diverse per scegliere diverse funzioni
- **Cicli:** utilizzati per ripetere il codice

Metteremo tutto insieme nel nostro codice **todo.py**. Iniziamo.

Inizia aprendo Thonny IDE (menu Raspberry Pi > Programmazione > Thonny). Possiamo scrivere i programmi con qualsiasi editor di testo, ma Thonny è molto amichevole per i principianti e ci permette di vederne gli elementi più in dettaglio.

> Python è potente abbastanza per i data scientist ed esperti di IA

È tradizione partire con qualsiasi nuovo linguaggio di programmazione visualizzando la riga di testo “Ciao, Mondo!". Immetti il seguente codice nel riquadro Codice nella metà superiore di Thonny

```
print("Ciao Mondo!")
```

Dovremmo salvarlo. Fai clic su Salva e assegnagli il nome **hello_world.py**. L'estensione ".py". indica è un programma Python e per convenzione, tutti i nomi dei file Python sono in minuscolo con parole separate da trattini bassi. Fai clic su Esegui e le parole "Ciao Mondo!" appariranno nella shell sottostante. Congratulazioni per aver eseguito il tuo primo programma!

Tutto è variabile

Un concetto chiave da apprendere sono le variabili (a volte chiamate "vars" in breve). Sono contenitori di archiviazione nel programma utilizzati per salvare (e riutilizzare) oggetti. Questi possono essere di diversi tipi: tipicamente numeri, parole e istruzioni booleane. I numeri sono conosciuti come “interi” o “int” se sono numeri tondi o "float" se hanno dei decimali. E le parole sono chiamate “stringhe” in linguaggio informatico. Gli operatori booleani sono dichiarazioni "Vero/Falso" utilizzate nel processo decisionale.

Crea un nuovo programma chiamato **vars.py** ed inserisci le seguenti righe:

L'Editor del Codice è dove ti trovi per inserire il codice. Premi Esegui nel Menu Strumenti per avviare il programma

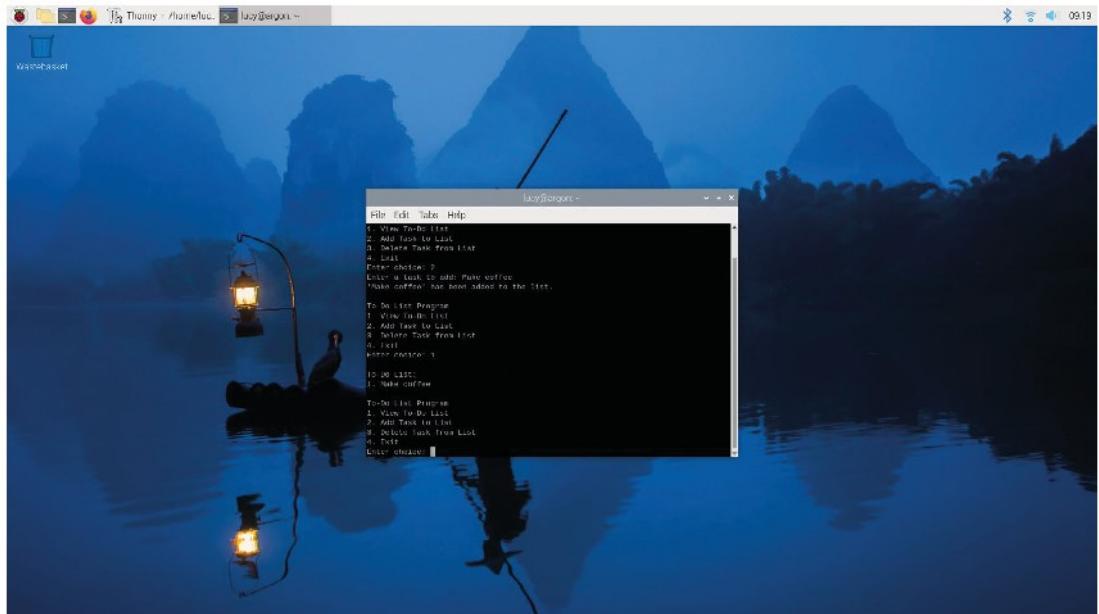
La Shell nella parte inferiore della interfaccia, visualizza l'output del codice. Qui digiti anche i comandi per interagire con il programma

```
name = "John" # String
age = 30 # Integer
height = 5.9 # Float
student = True # Boolean
print(f"Nome: {name}, Età: {age}, Altezza:
{height}m, Studente: {student}")
```

Questo programma ha quattro diversi tipi di variabili (indicate nei commenti). Nota che la nostra istruzione print qui è più complessa di Ciao Mondo. Inizia con `print(f` che indica che questa è una "stringa letterale formattata" nota come "f-string". La f stessa non è stampata, ma Python ora sa che le parole all'interno delle parentesi graffe (come `{name}` fanno riferimento a una variabile e inserisce il suo valore nell'output (quindi `{name}` diventa 'john').

Sintassi e indentazione

Quando si scrive un programma è importante comprenderne la sintassi. Questa è la struttura del codice. Assicurati di inserire il codice esattamente come lo vedi e troviamo che sia utile leggere i nostri programmi dalla prima riga all'ultima per verificare eventuali errori. Se ricevi errori, leggi attentamente e risolvi eventuali problemi.

La prossima cosa che vedremo è chiamata indentazione. I programmi Python hanno spesso alcune righe rientrate con spazi iniziali. Non sono solo estetici, ma indicano una relazione tra una riga di codice e le righe seguenti.

In Python, sono sempre blocchi di quattro spazi. Premi sempre il tasto **SPAZIO** quattro volte, non utilizzare **TAB**. Se hai più di un

Top Tip

Sii preciso

Durante la digitazione assicurati di digitare il codice esattamente come lo vedi. Usa gli stessi spazi e lettere minuscole/ maiuscole.

I programmi Python possono anche essere eseguiti direttamente in una finestra del Terminale

> Quando si scrive un programma è importante comprendere la sintassi

livello di indentazione premi il tasto **SPAZIO** 8 volte, 12 volte e così via (è generalmente considerato negativo avere troppe indentazioni poiché diventa difficile da capire).

Il rientro viene utilizzato per indicare istruzioni condizionali, cicli e per definire funzioni. Cominciamo con una dichiarazione condizionale.

A condizione che...

Un'istruzione condizionale è spesso chiamata if/ else (c'è anche un terzo componente: else-if). L'idea di base è questa. Se la prima riga è Vera, allora esegui il codice riportato qui di seguito rientrato di quattro righe:

```
if condition:
    # codice da eseguire se la condizione è vera
```

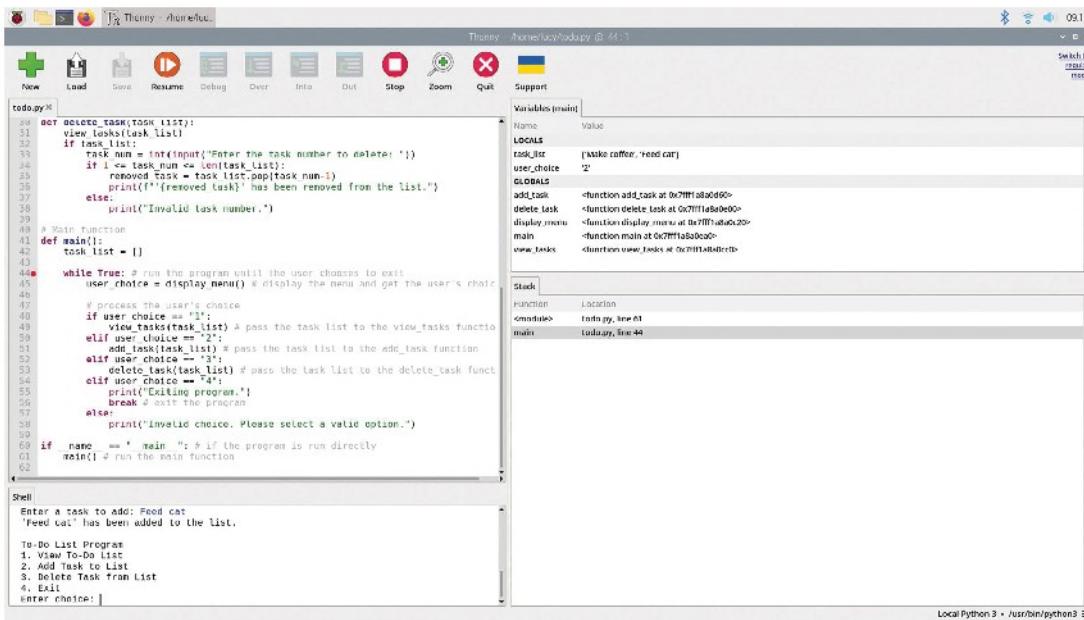
In genere le istruzioni if/else avranno opzioni alternative. Quindi, `if condition1` è Vera esegui la riga rientrata, `else` esegui invece il codice rientrato di seguito.

```
if condition1:
    # codice da eseguire se la condizione è vera
else:
    # codice da eseguire se la condizione è falsa
```

Mettiamolo in pratica:

```
name = "John" # Stringa
student = True # Booleana

if (student == True):
    print(f"Nome: {name} è uno studente!")
else:
    print(f"Name: {name} non è uno studente!")
```

Esegui questo codice e vedrai "Nome: John è uno studente!" La parte importante qui è che il codice sotto l'istruzione "else" non viene eseguito. Il rientro è il modo in cui vengono prese le decisioni nei programmi Python. Prova a cambiare True in False sulla riga 2 ed esegui nuovamente il programma.

Nota il simbolo `==` in (`student == True`). È "l'operatore di uguaglianza" che controlla se l'espressione restituisce True.

È diverso dal simbolo = in `name = "John"` e `student = True`. Questo singolo segno = è "l'operatore di assegnazione" che assegna il valore 'John' e True alle variabili.

Top Tip

PEP Style Guide

Leggi e aggiungi ai segnalibri la guida di stile Python **magpi.cc/pep8**

Utilizzare Debug invece di Esegui ti consente di indagare il codice e guardare le variabili

> L'indentazione viene utilizzata per indicare le istruzioni condizionali, cicli e definizione delle funzioni

È facile confondere le due cose, quindi usale con attenzione.

Creiamo un albero decisionale più complesso utilizzando le istruzioni **if**, **elif** e **else** e un operatore di confronto chiamato "minore di" indicato dal simbolo **<**.

```
name = "John" # Stringa
age = 23 # Intero

if (age < 16):
    print(f"Nome: {name} è sotto i 16!")
elif (age < 25):
    print(f"Nome: {name} è sotto i 25!")
else:
    print(f"Nome: {name} è oltre i 25!")
```

Qui ogni riga controlla se l'età è inferiore a un valore. E valutando Vero il risultato del confronto se il numero intero nella variabile età è inferiore al numero intero nella dichiarazione condizionale.

Gioca con la variabile età per eseguire diverse opzioni.

Cicla i cicli

I cicli hanno una struttura simile alle istruzioni if/else e sono anch'essi rientrati di quattro righe. In un ciclo, tuttavia, il codice rientrato è progettato per essere eseguito più volte in base a una condizione. Ci sono diversi tipi di loop: il nostro programma To-Do presenta un ciclo "while" che si ripete indefinitamente finché non viene soddisfatta una condizione:

```
while condizione:
    # Blocco di codice da eseguire
```

Creiamo un semplice programma per il conto alla rovescia da 10 a 1 e poi dice "Decollo!".

```
counter = 10 # inizializza il contatore
while counter > 0: # il ciclo esiste fino a
quando counter è maggiore di 0
    print(counter)
    counter -= 1 # decrementa il contatore
print("Decollo!")
```

Il nostro programma To-Do usa una condizione **while True** con una istruzione **break**. Questo esegue il programma finché non scegliamo di uscire. Quando usi **while True** fai attenzione a consentire al programma un modo di uscire, altrimenti crei un ciclo infinito, che sembra bello ma è generalmente considerato una pessima forma.

Eseguire funzioni

Il terzo tipo di codice indentato in Python è la funzione. È un blocco di codice riutilizzabile che può essere eseguire ancora e ancora semplicemente scrivendo il nome della funzione. Pensalo come un copia e incolla del codice.

Top Tip

Spazi, non tab

Usane sempre quattro spazi per rientrare il tuo codice invece di premere **TAB**. Si discute su questo, ma i principianti dovrebbero sempre iniziare con gli spazi.

```
# Simple To-Do Program in Python

# Displays the menu
# Returns the user's choice
def display_menu():
    print("\nTo-Do List Program")
    print("1. View To-Do List")
    print("2. Add Task to List")
    print("3. Delete Task from List")
    print("4. Exit")
    choice = input("Enter choice: ")
    return choice

# Displays the tasks in the list
def view_tasks(task_list):
    if not task_list: # if the list is empty
        print("Your to-do list is empty.")
    else:
        print("\nTo-Do List:")
        for index, task in enumerate(task_list, start=1):
            print(f"{index}. {task}") # print the task number and the task

# Adds a task to the list
def add_task(task_list):
    task = input("Enter a task to add: ")
    task_list.append(task)
    print(f"'{task}' has been added to the list.")

# Deletes a task from the list
def delete_task(task_list):
    view_tasks(task_list)
    if task_list:
        task_num = int(input("Enter the task number to delete: "))
```

```
>>> %Run todo.py

To-Do List Program
1. View To-Do List
2. Add Task to List
3. Delete Task from List
4. Exit
Enter choice:
```

▲ Thonny è un IDE (ambiente di sviluppo integrato) semplice da usare, installato di default in Raspberry Pi OS

Una funzione viene definita utilizzando **`def`**:

```
def nome_funzione(parameteri):
    # Blocco di codice
```

Puoi chiamare la funzione usando il comando **`nome_funzione`** e aggiungi eventuali elementi opzionali tra parentesi. Hai già visto una funzione incorporata in Python: **`print()`**.

Quando prima hai immesso **`print(“Ciao, Mondo!”)`** hai usato **`print()`** come comando funzione passando “Ciao, Mondo!” come argomento nei parametri. La funzione print ha poi prodotto “Ciao, Mondo!” come stringa a schermo.

Il comando **`def`** ti permette di creare le tue funzioni su misura. Creiamone una che saluti la persona passato nel parametro:

```
def greet(name): # name è il parametro
    print(f"Ciao, {name}!")
message = greet("Alice") # Alice è un argomento
message = greet("John") # John è un argomento
```

Esegui questo codice e vedrai "Ciao, Alice!” e “Ciao, John!”. Sta eseguendo lo stesso comando **`print()`** ogni volta ma con un argomento diverso.

Ricapitolando

C'è molto di più in Python negli altri linguaggi di programmazione, ma fondamentalmente tutto si condensa in funzioni, cicli e condizioni. Rieseguire il codice ancora e ancora, in modo leggermente diverso e prendendo decisioni diverse sulla base di ciò che sta accadendo.

Ci auguriamo che il viaggio ti piaccia. Amiamo la programmazione e ci aiuta a pensare, e risolvere i problemi. Questi sono i mattoni fondamentali della programmazione e ora che puoi identificarli la programmazione Python inizierà ad aprirsi.

Inserisci attentamente ogni riga del codice **todo.py** e ragiona su cosa fa ogni riga per il programma. Eseguilo e se ricevi un errore leggi il tuo codice attentamente e sistema la riga errata. Quando lo hai funzionante, giocarci.

Commenti

I commenti in Python vengono utilizzati per spiegare cosa dovrebbe fare il codice, rendendo il tutto più semplice per comprendere il codice a colpo d'occhio. L'interprete Python non esegue i commenti. Sono utilizzati esclusivamente a scopo di documentazione.

È buona norma aggiungere commenti per spiegare il tuo codice. Anche se servono solo a spiegarlo a te stesso.

I commenti in Python sono indicati con un simbolo # e quel che lo segue, viene ignorato:

```
# Questo è un commento a linea singola
print("Ciao Mondo!")  # Anche questo è un
commento, seguendo il codice sulla stessa riga
```

Non lesinare sui commenti. Sono un segno di buon codice ed è meglio abituarsi presto a usarli.
magpi.cc/pepcomments

todo.py

SCARICA IL CODICE COMPLETO:

magpi.cc/github

```
001.  # Semplice programma lista To-Do in Pthon
002.
003.  # Mostra il menu
004.  # Ritorna la scelta dell utente
005.  def display_menu():
006.      print("\nProgramma Lista To-Do")
007.      print("1. Vedi la Lista To-Do")
008.      print("2. Aggiungi voce alla Lista")
009.      print("3. Cancella voce dalla Lista")
010.      print("4. Esci")
011.      choice = input("Immetti la scelta: ")
012.      return choice
013.
014.  # Mostra i compiti della lista
015.  def view_tasks(task_list):
016.      if not task_list: # se la lista è vuota
017.          print("La tua Lista è vuota.")
018.      else:
019.          print("\nLista To-Do:")
020.          for index, task in enumerate(
      task_list, start=1):
021.              print(f"{index}. {task}") # fai print
      del numero e del compito
022.
023.  # Aggiungere voci alla lista
024.  def add_task(task_list):
025.      task = input("Immetti la voce da aggiungere: ")
026.      task_list.append(task)
027.      print(f"'{task}' è stato aggiunto alla
      lista.")
028.
029.  # Cancellare voci dalla lista
030.  def delete_task(task_list):
031.      view_tasks(task_list)
032.      if task_list:
033.          task_num = int(input(
      "Immetti il numero della voce da cancellare: "))
034.          if 1 <= task_num <= len(task_list):
035.              removed_task = task_list.pop(
      task_num-1)
036.              print(f"'{removed_task}' è stato
      rimosso dalla lista.")
037.          else:
038.              print("Numero non valido.")
039.
040.  # Funzione principale
041.  def main():
042.      task_list = []
043.
044.      while True: # esegui il programma fino a che
      non viene scelto di uscire
045.          user_choice = display_menu() # mostra il
      menu e recepisce la scelta dell' utente
046.
047.          # processa la scelta dell'utente
048.          if user_choice == "1":
049.              view_tasks(task_list) # passa la
      lista alla funzione view_tasks
050.          elif user_choice == "2":
051.              add_task(task_list) # passa la
      lista alla funzione add_task
052.          elif user_choice == "3":
053.              delete_task(task_list) # passa la
      lista alla funzione delete_task
054.          elif user_choice == "4":
055.              print("Uscita dal programma.")
056.              break # esci dal programma
057.          else:
058.              print("Scelta non valida. Cortesemente, scegli
      una opzione valida.")
059.
060.  if __name__ == "__main__": # se il programma viene
      eseguito direttamente
061.      main() # esegui la funzione main
```

Parte 02

Conosci il tuo Raspberry Pi

Scopri gli elementi essenziali che serviranno al tuo Raspberry Pi, e scopri come collegarli tutti per configurarlo e farlo funzionare

MAKER

Gareth Halfacree

Gareth è un giornalista tech freelance, scrittore ed ex amministratore di sistema nel settore dell'istruzione con una passione per software e hardware open source.

freelance.halfacree.co.uk

Attenzione! Alimentazione

Raspberry Pi 5 necessita di un alimentatore da 5 V in grado di erogare 5A di corrente, e un cavo USB-C. Se colleghi un alimentatore con una corrente inferiore, compreso l'alimentatore ufficiale per Raspberry Pi 4, le porte USB del Pi 5 saranno limitate solo a dispositivi a bassa potenza.

magpi.cc/power

Raspberry Pi è progettato per essere il più possibile facile e veloce da configurare e utilizzare, ma, come qualsiasi computer, si basa su vari componenti esterni chiamati periferiche. Sebbene sia facile dare un'occhiata al circuito stampato nudo di Raspberry Pi – che appare significativamente diverso dai computer racchiusi a cui potresti essere abituato – e preoccuparsi che le cose stiano per complicarsi, non è così. Puoi mettere in funzione il tuo Raspberry Pi in ben meno di dieci minuti se segui i passaggi di questa guida.

Se hai acquistato un kit desktop Raspberry Pi o un Raspberry Pi 400, avrai già quasi tutto ciò che serve per iniziare. Tutto quel che necessiterà sarà un monitor da computer o una TV con porta HDMI – lo stesso tipo di connettore utilizzato da set-top box, lettori Blu-ray e console di gioco – così puoi vedere cosa sta facendo il tuo Raspberry Pi.

Se hai preso il tuo Raspberry Pi senza accessori, allora avrai bisogno anche di:

1. Alimentatore USB. Da 5 V nominali e 5 A con un connettore USB-C per Raspberry Pi 5, un alimentatore da 5 V e 3 A con USB-C per Raspberry Pi 4 Modello B o Raspberry Pi 400, oppure un alimentatore da 5 V con potenza nominale di 2,5 A con un connettore micro-USB per Raspberry Pi Zero 2 W. Sono consigliati gli alimentatori ufficiali Raspberry Pi (**Figura 1**), che sono progettati per far fronte alle rapide variazioni di richieste di potenza di Raspberry Pi. Gli alimentatori di terze parti potrebbero non essere in grado di farlo e potrebbero causare problemi al tuo Raspberry Pi.

2. Scheda microSD **(Figura 2)** funge da memoria permanente del tuo Raspberry Pi. Tutti i file che crei e tutto il software che installi, insieme al sistema operativo stesso, sono memorizzati sulla scheda. È sufficiente una scheda da 8 GB per iniziare, anche se una da 16 GB offre più spazio per il futuro. Il kit desktop Raspberry Pi include una scheda microSD con Raspberry Pi OS preinstallato.

3. Tastiera e mouse USB (**Figura 3)** ti consentono di controllare il tuo Raspberry Pi. Quasi tutti i mouse e tastiere cablati o wireless con connettore USB funzioneranno con Raspberry Pi, anche se qualche tastiera gaming con luci colorate può risultare troppo esosa di corrente per essere utilizzata in modo affidabile. Raspberry Pi Zero 2 W necessita di un adattatore micro-USB OTG, e se desideri collegare più di un dispositivo USB alla volta, avrai bisogno di un hub USB alimentato.

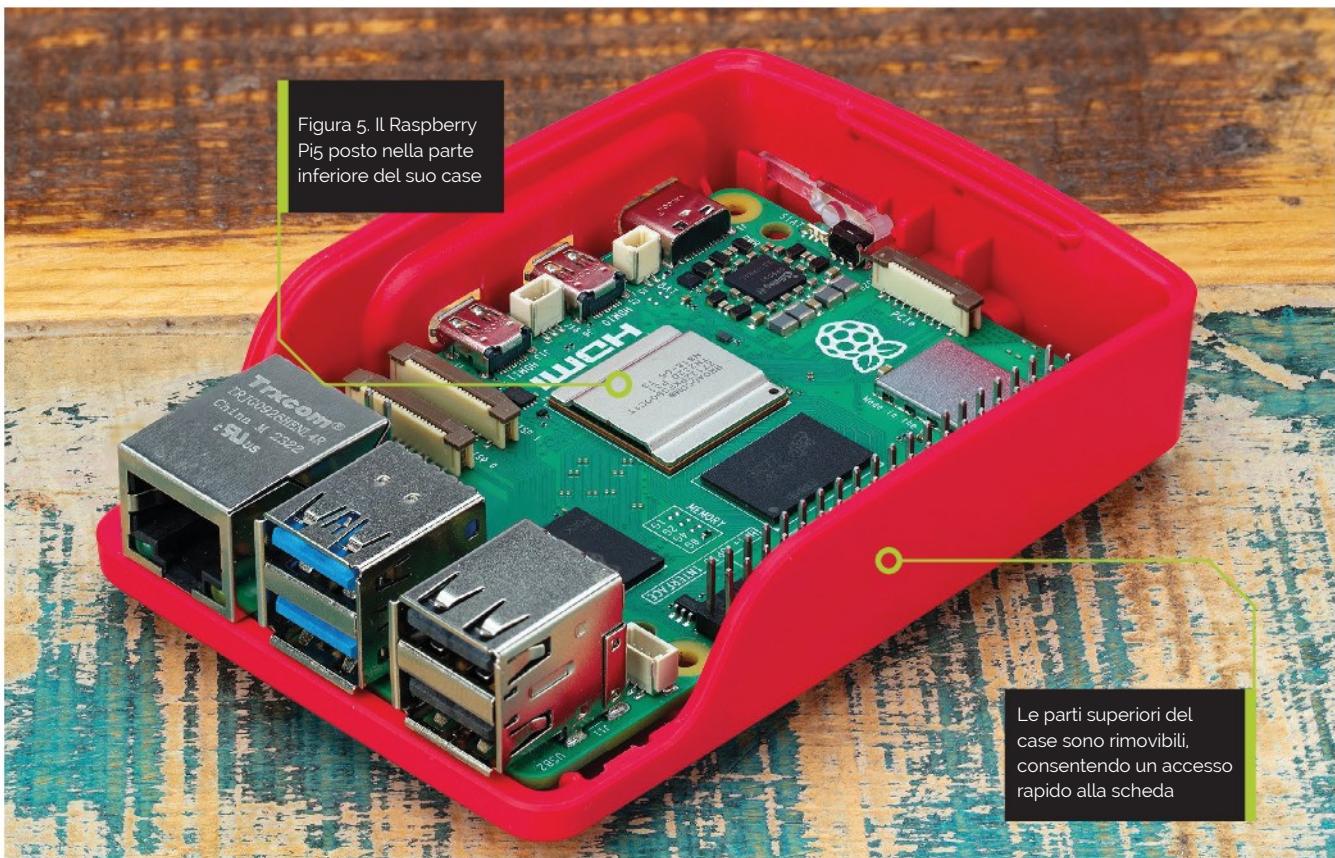



Figura 2: scheda microSD

Figura: 1 alimentatore USB

Figura 5. Il Raspberry Pi5 posto nella parte inferiore del suo case

Le parti superiori del case sono rimovibili, consentendo un accesso rapido alla scheda

4. Cavo HDMI. Trasporta suoni e immagini dal tuo Raspberry Pi alla tua TV o monitor. Raspberry Pi 4, Raspberry Pi 5 e Raspberry Pi 400 necessitano di un cavo con connettore micro-HDMI ad un'estremità (**Figura 4**), mentre Raspberry Pi Zero 2 W necessita di un cavo con connettore mini-HDMI; l'altra estremità dovrebbe avere un connettore HDMI full-size per il display. Puoi anche usare un adattatore da micro o mini-HDMI a HDMI con un cavo HDMI standard. Se utilizzi un monitor senza presa HDMI, puoi acquistare adattatori per convertirla in connettori DVI-D, DisplayPort o VGA.

▲ **Figura 3:** tastiera USB

▶ **Figura 4:** cavo HDMI

Raspberry Pi è sicuro da usare senza custodia, a patto di non posizionarlo su una superficie metallica, che potrebbe condurre elettricità e provocare un cortocircuito. Un case facoltativo, tuttavia, può fornire protezione aggiuntiva; il kit desktop include il case ufficiale Raspberry Pi, mentre custodie di terze parti sono disponibili presso tutti i buoni rivenditori.

Se desideri utilizzare Raspberry Pi 4, Raspberry Pi 5 o Raspberry Pi 400 con una rete cablata, invece che con una rete Wi-Fi, ti servirà anche un cavo di rete Ethernet. Questo dovrebbe essere collegato a uno switch o al router della tua rete. Se stai progettando di utilizzare il wireless integrato di Raspberry Pi, non servirà un cavo; dovrai tuttavia conoscere il nome e la password o passphrase della tua rete wireless.

Impostare l'hardware

Inizia togliendo il Raspberry Pi dalla sua scatola. Raspberry Pi è un componente hardware robusto, ma ciò non significa che sia indistruttibile; prova a prendere l'abitudine di tenere la scheda per i bordi, anziché sui lati piatti, e a fare extra attenzione ai pin metallici che sporgono. Se questi pin si piegano, nella migliore delle ipotesi sarà difficile utilizzare schede add-on e altro hardware aggiuntivo,

Cosa Serve

- Raspberry Pi
- Raspberry Pi OS

Top Tip

Impostare Raspberry Pi 400

queste istruzioni sono per l'impostazione di Raspberry Pi 5 o altra scheda nuda della famiglia Raspberry Pi. Per Raspberry Pi 400, vedi le istruzioni di impostazioni specifiche per Raspberry Pi 400.

Figure 6: Connessione del connettore della ventola

nel peggiore dei casi, potrebbe causare un cortocircuito che potrebbe danneggiare il tuo Raspberry Pi.

Se non l'hai già fatto, dai un'occhiata a *Conosci il tuo Raspberry Pi* nel numero 139 di The MagPi (**bit.ly/MagPi139It**), per i dettagli su dove sono esattamente le varie porte e cosa fanno.

Assemblare il case Raspberry Pi

Se stai installando Raspberry Pi 5 in una case, questo dovrebbe essere il tuo primo passo. Se stai utilizzando il case ufficiale Raspberry Pi, inizia dividendolo nei suoi tre pezzi: base rossa, telaio e supporto ventola e coperchio bianco.

Prendi la base e tienila in modo che l'estremità rialzata sia alla tua sinistra e l'estremità ribassata alla tua destra.

Tieni il tuo Raspberry Pi 5, senza scheda microSD, dalle sue porte USB ed Ethernet, un po' angolato. Abbassa delicatamente l'altro lato nella base, come in **Figura 5**. Dovresti sentire uno scatto mentre lo appoggi contro la base.

Sistemare la ventola

La ventola dovrebbe essere arrivata inserita nel suo supporto, e questo dovrebbe già essere inserito nel suo telaio quando lo estrai dalla scatola. In caso contrario, puoi incastrarli tra loro.

Quindi, collega il connettore JST bianco della ventola al connettore dedicato sul Raspberry Pi 5 come mostrato in **Figura 6**. Entra solo in un modo, quindi non ti devi preoccupare di collegarlo al contrario.

Incastra il gruppo ventola e telaio in posizione come mostrato nella **Figura 7** e spingi delicatamente verso il basso fino a quando non sentirai un clic.

Se vuoi racchiudere tutto nella custodia, prendi il coperchio bianco opzionale e posizionalo in modo che il logo Raspberry Pi sia sopra le porte USB ed Ethernet su Raspberry Pi 5, come mostrato in **Figura 8**. Per fissarlo in posizione, spingi il centro del coperchio fino allo scatto.

Hat e coperchi

È possibile montare un HAT (Hardware on Top) direttamente sopra Raspberry Pi 5 rimuovendo il gruppo ventola oppure è possibile impilarlo sopra la ventola e il telaio utilizzando distanziatori alti 14 mm e un estensore GPIO da 19 mm. Questi saranno disponibili separatamente dai rivenditori autorizzati.

Assemblare il case di Raspberry Pi Zero

Se stai installando Raspberry Pi Zero 2 W in un case, questo dovrebbe essere il tuo primo passo. Se stai utilizzando il case ufficiale Raspberry Pi Zero, inizia smontandolo. Dovresti avere quattro pezzi: una base rossa e tre coperchi bianchi.

Se stai utilizzando Raspberry Pi Zero 2, vorrai utilizzare il coperchio solido. Se utilizzerai il GPIO, scegli il coperchio con il lungo foro

Figura 7: Attaccare il supporto ventola e il telaio

Figura 8: Mettere il coperchio sopra il case

▲ **Figura 9:** Il case Raspberry Pi Zero

▲ **Figura 10:** Mettere lo Zero nel suo case

> La scheda microSD scivolerà dentro, fermati senza il clic

rettangolare. Se hai un Camera Module 1 o 2, scegli il coperchio con il foro circolare.

Il Camera Module 3 e il l'HQ Camera non sono compatibili con Il coperchio per fotocamera del case Raspberry Pi Zero e devono essere utilizzati fuori custodia; c'è una fessura all'estremità del case Raspberry Pi Zero per il cavo piatto della fotocamera.

Prendi la base e posizionala piatta sul tavolo in modo che le aperture per le porte siano rivolte verso di te, come mostrato nella **Figura 9**.

Tenendo il tuo Raspberry Pi Zero (con il scheda microSD inserita) dai bordi, allinealo in modo che i piccoli pioli circolari negli angoli della base entrino nei fori di montaggio negli angoli del circuito stampato del Raspberry Pi Zero 2 W. Quando sono allineati (**Figura 10**), spingi delicatamente Raspberry Pi Zero 2 W verso il basso finché non senti un clic e le porte saranno allineate con le aperture nella base.

Prendi il coperchio bianco che hai scelto e posizionalo sopra alla base del case, come mostrato nel Capitolo 11. Se stai utilizzando il coperchio per il Camera Module, assicurati che il cavo non sia schiacciato. Quando il coperchio è a posto, spingilo delicatamente verso il basso finché non senti un clic.

▲ **Figura 12:** Attaccare i piedini

Camera Module con il case dello Zero

Se utilizzi un Camera Module Raspberry Pi, usa il coperchio con il foro circolare. Allinea i fori di montaggio del Camera Module con i supporti a forma a croce nel coperchio, in modo che il connettore della fotocamera sia rivolto verso il logo sul coperchio. Incastralo in posizione. Alza delicatamente la linguetta sul connettore della fotocamera, quindi inserisci l'estremità più stretta del cavo a nastro incluso nel connettore, prima di reinserire la linguetta al suo posto. Collega l'estremità più larga del cavo al Camera Module, nello stesso modo.

A questo punto puoi anche incollare i piedini in gomma inclusi, sul fondo del case (vedi **Figura 12**): giralo, stacca i piedini dal foglio di supporto e incollali nelle rientranze circolari della base per fornire un miglior grip sulla scrivania.

Collegamento della scheda microSD

Per installare la scheda microSD, che è lo spazio di archiviazione di Raspberry Pi, capovolgi il tuo Raspberry Pi (nella sua custodia se ne stai usando una) e fai scorrere la schedina nello slot microSD con il lato dell'etichetta verso l'alto. Entrerà soltanto in un modo, e dovrebbe scivolare in sede senza troppa pressione (vedere **Figura 13**).

La scheda microSD scivolerà dentro, quindi fermati senza clic.

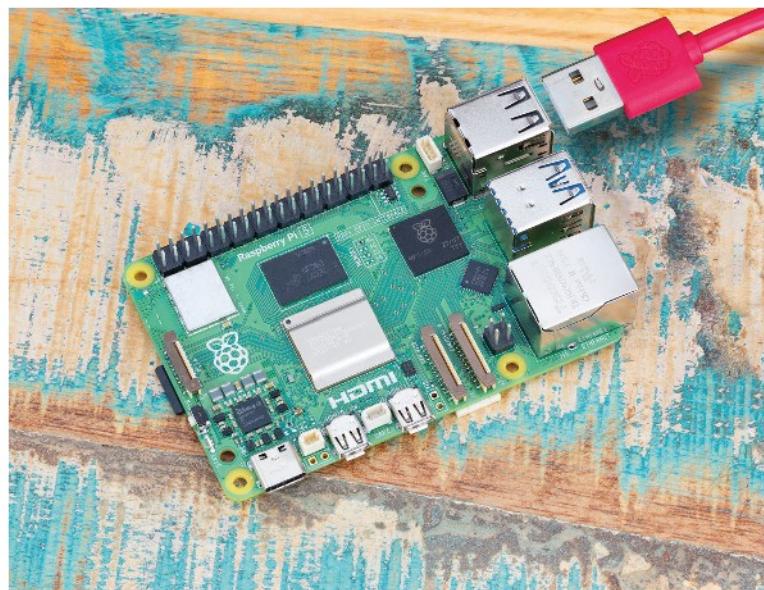
Per Raspberry Pi Zero 2 W, lo slot microSD è in alto a sinistra. Inserisci la scheda con l'etichetta rivolta in su.

Se desideri rimuoverla in futuro, afferra semplicemente l'estremità della schedina e tirala delicatamente fuori. Se stai utilizzando un modello precedente di Raspberry Pi, dovrai prima dare una leggera spinta alla scheda per sbloccarla; questo non è necessario con un Raspberry Pi 3, 4, 5 o qualsiasi modello di Raspberry Pi Zero.

▼ **Figura 13:** Inserire la scheda microSD

Collegare tastiera e mouse

Collega il cavo USB della tastiera a una qualsiasi delle quattro porte USB (USB 2.0 nere o USB 3.0 blu) di Raspberry Pi come mostrato nella **Figura 14.** Se stai utilizzando la tastiera ufficiale Raspberry Pi, c'è una porta USB sul retro, per il mouse. Altrimenti, basta collegare il cavo USB del tuo mouse a un'altra porta USB di Raspberry Pi.

Per Raspberry Pi Zero 2 W, dovrai utilizzare un adattatore micro-USB OTG. Inseriscilo nella porta micro-USB sinistra, quindi collega ol cavo USB della tastiera all'adattatore USB OTG.

Se stai usando una tastiera e mouse separati, anziché una con touchpad integrato, dovrai anche utilizzare un hub USB alimentato. Collega l'adattatore micro-USB OTG come sopra, quindi collega il cavo USB dell'hub all'adattatore USB OTG prima di collegare la tastiera e il mouse all'hub USB. Infine, collega l'alimentatore dell'hub e accendilo.

I connettori USB per tastiera e mouse dovrebbero entrare in sede senza troppa pressione; Se devi forzare il connettore, allora qualcosa non va. Controlla che il connettore USB sia nel verso giusto!

Tastiera e mouse

Tastiera e mouse permettono di dire a Raspberry Pi cosa fare; nell'informatica, questi sono conosciuti come dispositivi di input, al contrario del display, che è un dispositivo di output.

Figura 14: Inserire un cavo USB nel Raspberry Pi 5

Connettere un display

Per Raspberry Pi 4 e Pi 5, prendi il cavo micro-HDMI e collega l'estremità più piccola alla porta micro-HDMI più vicina alla porta USB Type-C del tuo Raspberry Pi. Collega l'altra estremità al display, come mostrato nella **Figura 16**.

Per Raspberry Pi Zero 2 W (**Figura 15**), prendi il cavo mini-HDMI e collega l'estremità più piccola alla porta mini-HDMI sul lato sinistro del tuo Raspberry Pi, sotto lo slot microSD. L'altro capo va connesso al display.

Se il tuo display ha più di una porta HDMI, cerca un numero di porta accanto al connettore; dovrai commutare lo schermo su questo ingresso per vedere il display del tuo Raspberry Pi. Se non trovi il numero della porta, non preoccuparti, puoi semplicemente passare da un ingresso all'altro fino a trovare la schermata del Raspberry Pi.

Figura 15: Collegare il cavo HDMI a Raspberry Pi Zero

Figura 16: Collegare il cavo HDMI a Raspberry Pi 5

Figura 17: Connettere il Raspberry Pi 5 all'Ethernet

Collegare una TV

Se la tua TV o monitor non dispone di connessione HDMI, non significa che non puoi utilizzare un Raspberry Pi. Cavi adattatori, disponibili da qualsiasi rivenditore di elettronica, ti consentiranno di convertire la porta micro o mini HDMI sul tuo Raspberry Pi in DVI-D, DisplayPort o VGA per l'utilizzo con altri monitor per computer.

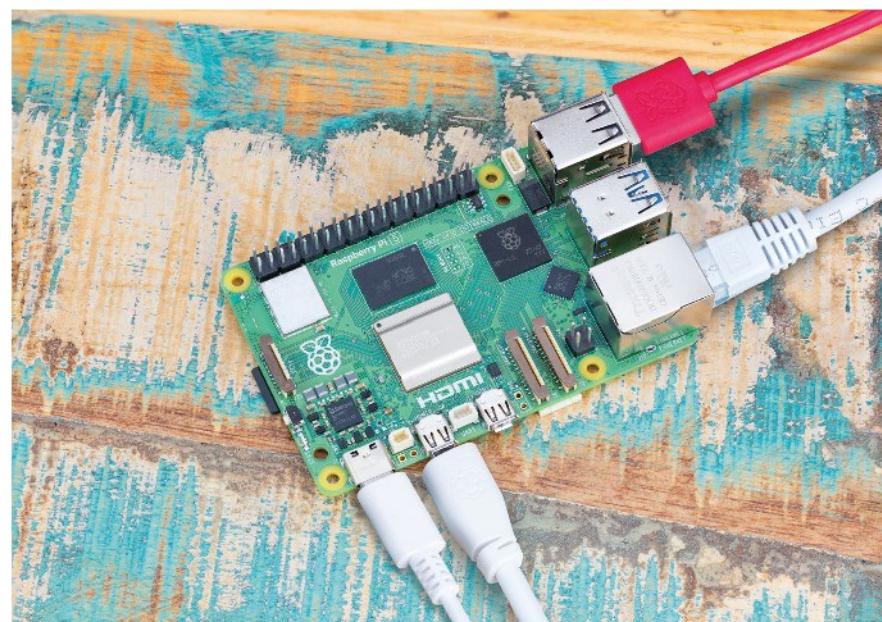
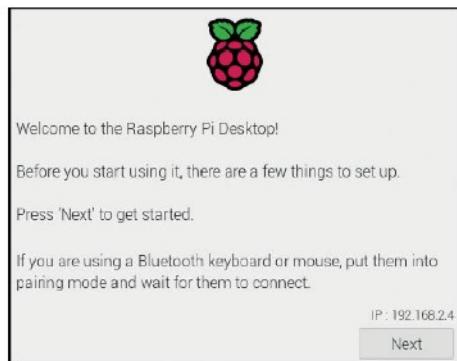
Connettere un cavo di rete (opzionale)

Per connettere il tuo Raspberry Pi a una rete cablata, prendi un cavo di rete e inserirlo nella porta Ethernet di Raspberry Pi, con la clip di plastica rivolta verso il basso, finché non si sente un clic (vedi **Figura 17**). Se è necessario rimuovere il cavo, premi la clip di plastica e estrai delicatamente la spinetta del cavo.

L'altra estremità del cavo di rete dovrebbe essere collegata a qualsiasi porta libera dell'hub di rete, switch o router, nello stesso modo.

Collegare un alimentatore

Collegare il tuo Raspberry Pi a un alimentatore è il passaggio finale nel processo di configurazione dell'hardware. l'ultima cosa che farai prima di iniziare la configurazione del software. Il tuo Raspberry Pi si accenderà non appena sarà collegato a un alimentatore.

Figura 18: Alimentare il Raspberry Pi 5

> Raspberry Pi si accenderà non appena collegato a un alimentatore

Per Raspberry Pi 4 e Raspberry Pi 5, connetti l'estremità più piccola del cavo USB-C di alimentazione al connettore USB-C sul tuo Raspberry Pi, come nella **Figura 18**. Può andare in entrambi i sensi e dovrebbe entrare dolcemente in sede. Se il tuo alimentatore ha un cavo staccabile, assicurati che l'altra estremità sia collegato al corpo dell'alimentatore.

Per Raspberry Pi Zero 2 W, collega l'estremità più piccola del cavo micro-USB di alimentazione alla porta micro-USB sul tuo Raspberry Pi. Entra in un solo verso, quindi fai attenzione a non forzare.

Congratulazioni: hai approntato il tuo Raspberry Pi (**Figura 19**)!

Figure 19: Il tuo Raspberry Pi è pronto!

Figura 20: La procedura guidata di benvenuto di Raspberry Pi OS

Infine, collega l'alimentatore alla rete elettrica e il tuo Raspberry Pi inizierà immediatamente a funzionare.

Vedrai brevemente un cubo color arcobaleno seguito da una schermata informativa con un Logo Raspberry Pi. Potresti anche vedere una schermata blu che viene visualizzata man mano che il sistema operativo viene ridimensionato per sfruttare appieno la scheda microSD. Se vedi una schermata nera, attendi qualche minuto; la prima volta che si avvia, Raspberry Pi eseguirà alcune pulizia in background, che possono richiedere un po' di tempo.

Dopo un po' vedrai la procedura guidata di benvenuto di Raspberry Pi OS (**Figura 20**). Il tuo OS ora è pronto per essere configurato.

Impostare Raspberry Pi 400

A differenza del Raspberry Pi 4, Raspberry Pi 400 han tastiera integrata e scheda microSD già installata. Dovrai comunque collegarne alcuni cavi per iniziare, ma dovrebbero richiedere solo pochi minuti.

Collegare un mouse

La tastiera del Raspberry Pi 400 è già collegata, devi solo aggiungere un mouse. Prendi il Cavo USB del mouse e inseriscilo in una qualsiasi delle tre porte USB (2.0 o 3.0) sul pannello

Figura 21: Inserire un cavo USB nel Raspberry Pi 400

Figura 22: Collegare il cavo HDMI a Raspberry Pi 400

Figura 23: Connettere Raspberry Pi 400 all'Ethernet

posteriore del tuo Raspberry Pi 400. Se vuoi tenere le due porte USB 3.0 blu ad alta velocità per altri accessori, utilizza la porta USB 2.0, bianca.

Anche il connettore USB dovrebbe andare in sede senza molta pressione (vedi **Figura 21**). Se devi forzare il connettore, c'è qualcosa che non va. Controlla che sia rivolto nel verso giusto!

Collegare un display

Prendi il cavo micro-HDMI e collega l'estremità più piccola alla porta micro-HDMI più vicina allo slot microSD di Raspberry Pi 400 e l'altra estremità, al display, come mostrato nella **Figura 22**. Se il tuo display ha più di una porta HDMI, cerca un numero di porta accanto al connettore stesso; dovrai poi commutare la TV o il monitor su questo ingresso per vedere il display di Raspberry Pi. Se non riesci a vedere il numero della porta, non preoccuparti: puoi semplicemente scorrere ogni ingresso finché non trovi il Raspberry Pi.

Collegare un cavo di rete (opzionale)

Per collegare il tuo Raspberry Pi 400 a una rete cablata, usa un cavo di rete Ethernet e inseriscilo nel Raspberry Pi 400, con la clip di plastica rivolta verso l'alto, fino a sentire uno scatto (**Figura 23**). Se dovrai rimuovere il cavo,

Figura 24: Il tuo Raspberry Pi 400 è completamente cablato

basta premere la linguetta di plastica e far scorrere la spinetta delicatamente per liberare il cavo. L'altra estremità del tuo cavo di rete deve essere collegata a qualsiasi porta libera del tuo hub di rete, switch o router, nello stesso modo.

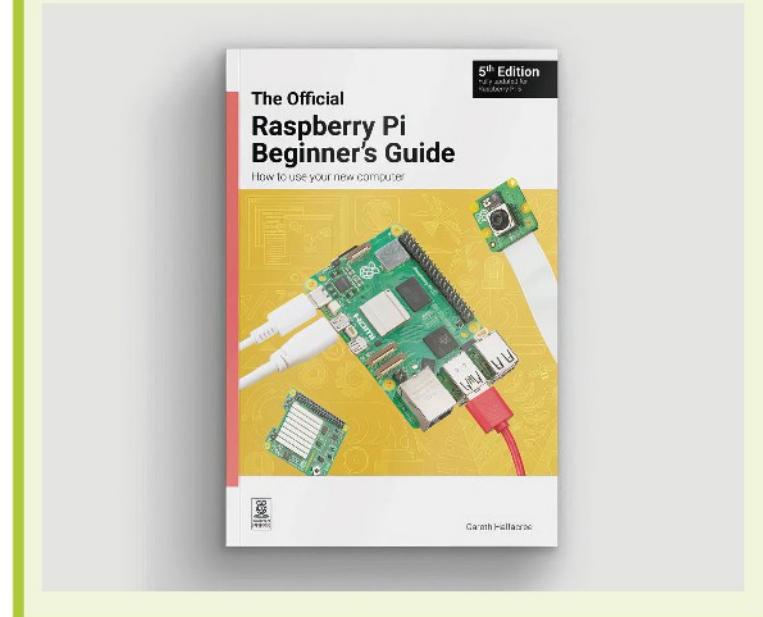
Collegare un alimentatore

Collegare il Raspberry Pi 400 a un alimentatore è l'ultimo passaggio del processo di configurazione dell'hardware, ed è una cosa che dovresti fare solo quando sei pronto a impostare il suo software. Raspberry Pi 400 non ha un interruttore di alimentazione e si accenderà non appena lo alimenti.

Innanzitutto, collega l'estremità USB Type-C del cavo di alimentazione al connettore USB Type-C di alimentazione su Raspberry Pi. Entra in entrambi i sensi e dovrebbe scivolare dolcemente in sede. Se il tuo alimentatore e ha un cavo staccabile, assicurati che l'altra estremità sia ben collegata al corpo dell'alimentatore.

Infine, collega l'alimentatore a una presa di corrente: il tuo Pi 400 inizierà immediatamente a funzionare. Congratulazioni, hai approntato il tuo Raspberry Pi 400 (**Figura 24**)!

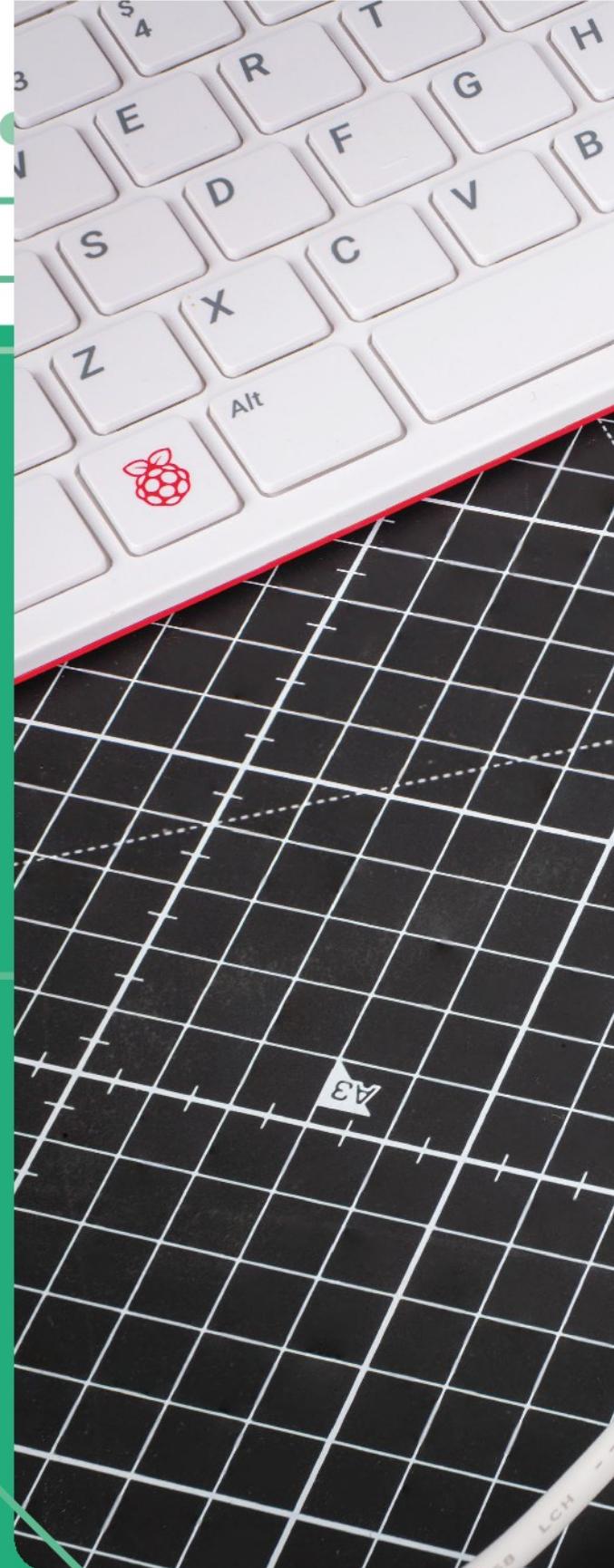
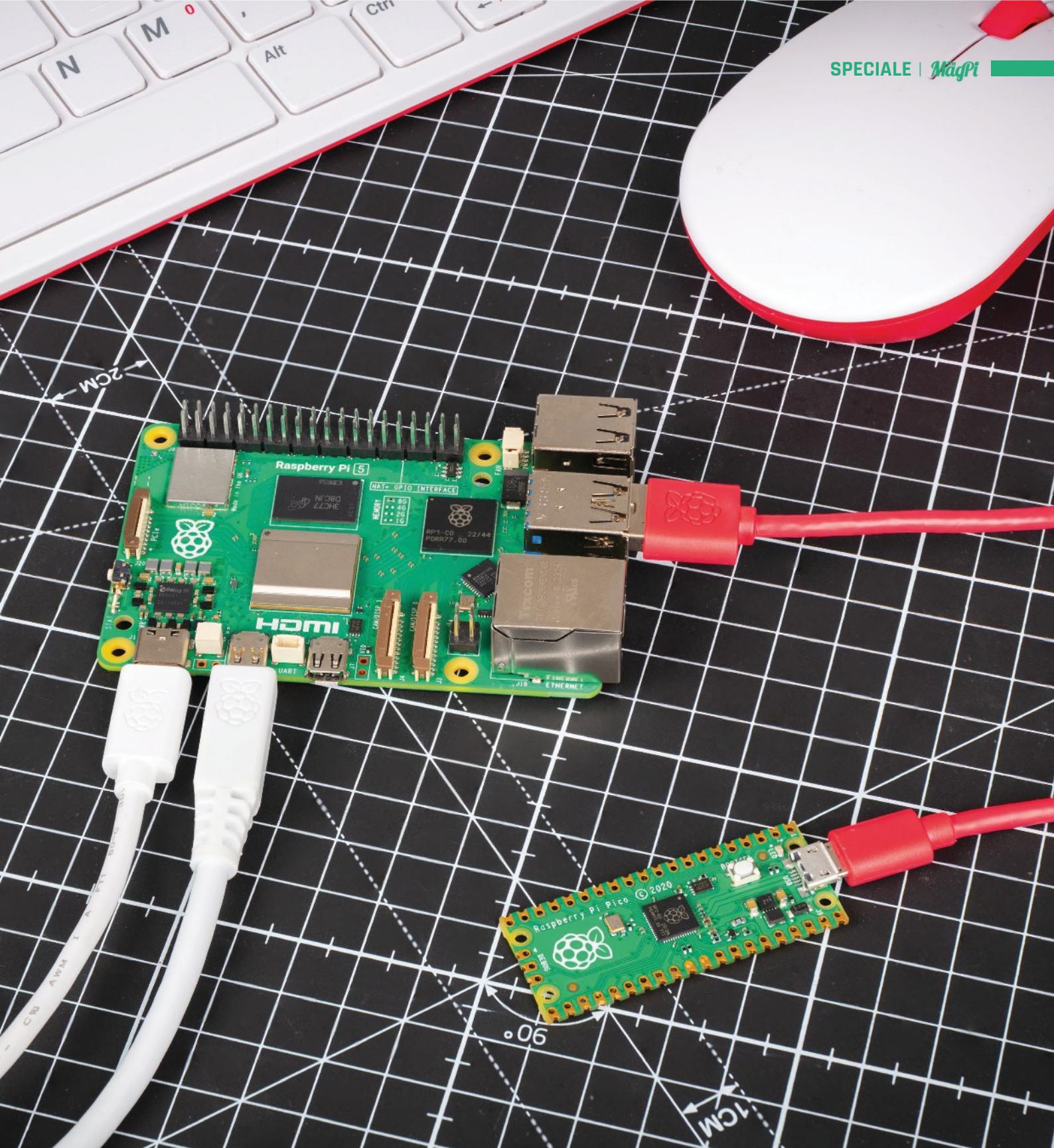
Vedrai brevemente un cubo arcobaleno seguito da una schermata informativa con il logo Raspberry Pi. Potresti anche vedere una schermata blu che viene visualizzata man mano che l'OS viene ridimensionato per sfruttare appieno la scheda microSD. Se vedi una schermata nera, attendi qualche minuto: la prima volta che si avvia, Raspberry Pi eseguirà alcune pulizie in background, che possono richiedere un po' di tempo.

Dopo un po' vedrai la procedura guidata di benvenuto di Raspberry Pi OS. Il tuo OS ora è pronto per essere configurato

La Guida per Principianti Ufficiale Raspberry Pi

Questo tutorial è tratto dalla quinta edizione di *The Official Raspberry Pi Beginner's Guide*. Questo libro di 290 pagine è ricco di indicazioni su come configurare il tuo Raspberry Pi, installare il suo sistema operativo e iniziare a utilizzare questo computer.

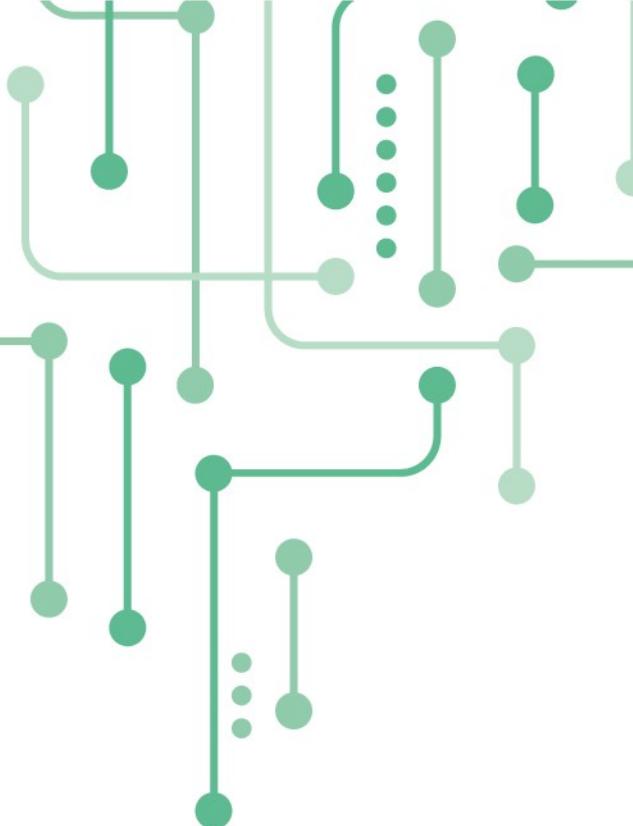
Con questo libro puoi iniziare a programmare progetti, utilizzando guide passo passo e i linguaggi di programmazione Scratch 3, Python e MicroPython.

Inoltre sperimenterai il collegamento di componenti elettronici e ti divertirai creando progetti straordinari. Puoi acquistarne una copia dal nostro negozio online o leggerla nell'app Bookshelf in Raspberry Pi OS.

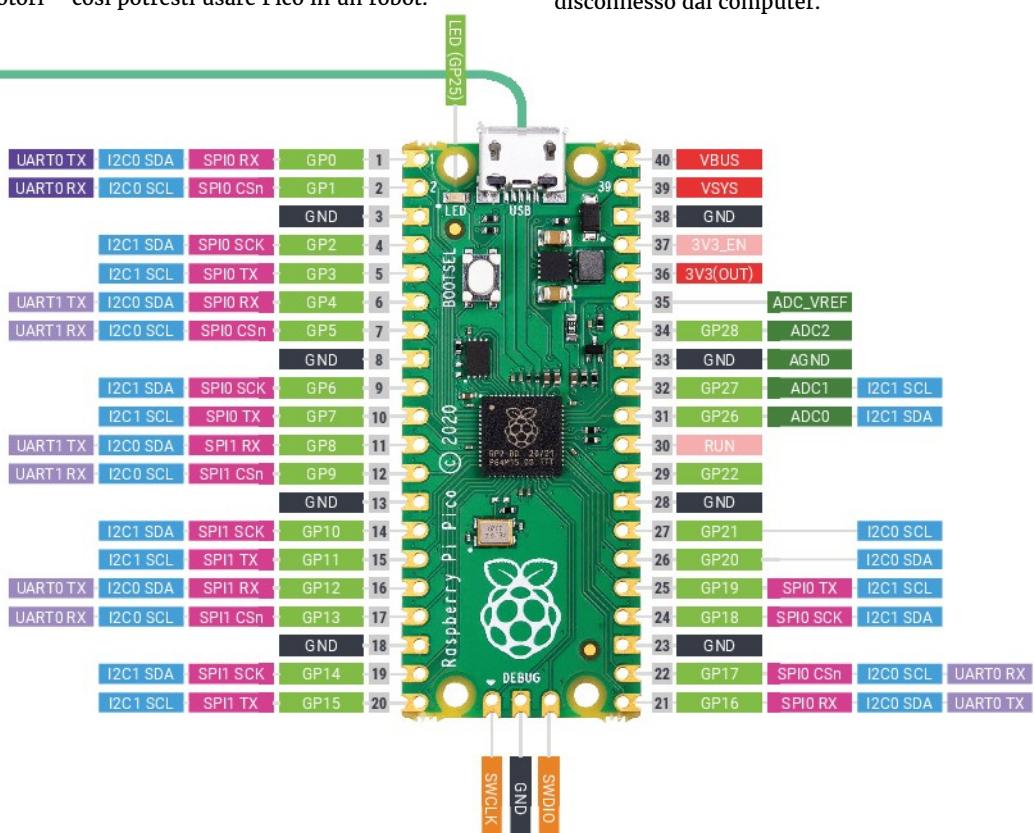
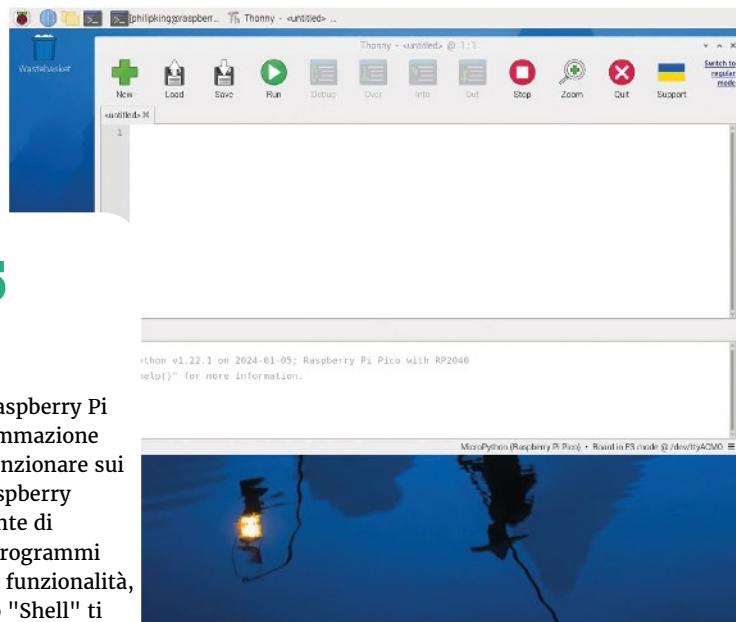
magpi.cc/beginnersguide

Raspberry Pi 5 & Raspberry Pi Pico

Di Phil King

Collegando Raspberry Pi 5 a Raspberry Pi Pico, ottieni il meglio da entrambi i mondi: un computer a scheda singola collegato a un microcontrollore compatto e a basso costo. Pico può essere programmato in linguaggio MicroPython utilizzando l'IDE Thonny su Raspberry Pi 5. È poi in grado di eseguire automaticamente un programma quando usato da solo, quindi può essere utilizzato in tutti i tipi di progetti portatili.

Qui ti mostreremo come configurare Pico con Raspberry Pi 5, installare su di esso il firmware MicroPython e scrivere un programma per far lampeggiare il LED a bordo. Daremo anche un'occhiata al Debug Probe che può essere utilizzato per vedere esattamente cosa sta facendo il processore del Pico.

Presentazione di **Raspberry Pi Pico**

Raspberry Pi Pico è una scheda microcontrollore compatta e economica in grado di eseguire programmi con un consumo energetico molto basso, il che lo rende ideale per progetti portatili.

Sebbene non sia potente come un computer a scheda singola Raspberry Pi, si basa su un chip microcontrollore RP2040 con due core Arm Cortex M0+ che funzionano fino a 133 MHz e macchine a stati PIO (Ingresso/Uscita Programmabili) che possono eseguire il codice indipendentemente dalla CPU principale. Pico dispone anche di 264kB di SRAM, 2MB di memoria flash integrata e i modelli W dispongono anche di Wi-Fi.

Proprio come qualsiasi altra scheda Raspberry Pi, Pico ha 40 pin GPIO, anche se qui son disposti in file su entrambi i lati della scheda (vedi il box "Pico Pinout" per i dettagli). Oltre ad permettere la connessione di una vasta gamma di schede di terze parti, è possibile utilizzare i pin GPIO per collegare i propri circuiti comprendenti componenti elettronici standard come LED, pulsanti, sensori e motori – così potresti usare Pico in un robot.

Pico può essere programmato da un computer collegato tramite USB, anche un Raspberry Pi, utilizzando il facile linguaggio MicroPython o, per gli utenti più avanzati, il C/C++.

Pico può poi eseguire automaticamente un programma all'accensione, quindi puoi usarlo disconnesso dal computer.

Raspberry Pi 5 e Thonny IDE

Il modo più semplice per programmare Raspberry Pi Pico è MicroPython, linguaggio di programmazione compatibile con Python sviluppato per funzionare sui microcontrollori. Con Pico collegato al Raspberry Pi 5 tramite USB, puoi usare l'IDE (ambiente di sviluppo integrato) Thonny per scrivere programmi MicroPython ed eseguirli su Pico. Ricco di funzionalità, Thonny è molto intuitivo e il suo riquadro "Shell" ti mostrerà tutti gli output dal programma Pico in esecuzione o i messaggi di errore, se qualcosa va storto. Ha anche un debugger integrato che ti consente di esplorare un programma passo dopo passo per mostrare quel che succede in ogni fase, aiutandoti a trovare i bug.

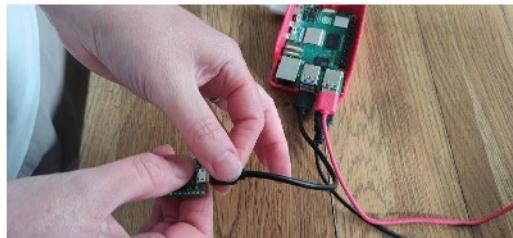
Debug Probe

Sebbene non sia essenziale per programmare Pico, soprattutto con MicroPython, questo add-on opzionale consente agli utenti avanzati di vedere esattamente cosa sta facendo Pico, con una console seriale. È particolarmente utile per il debug del codice C/C++ in esecuzione direttamente sul processore RP2040 di Pico.

Alimentato a sua volta da un RP2040, il Debug Probe ha tre connettori. La porta micro-USB è per collegarlo a un computer host, come Raspberry Pi 5, per gli strumenti di debug e visualizzare l'output. Ci sono anche due porte JST a tre pin, utilizzate per collegare i pin UART di Pico (vedi il box "Pinout Pico") e/o la porta SWD (Serial Wire Debug).

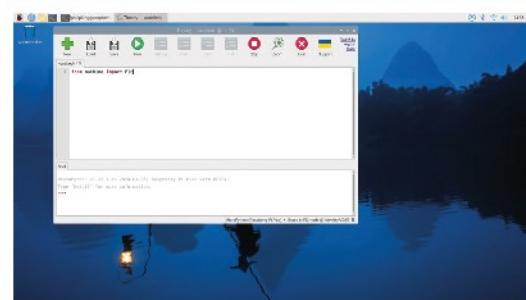
Top Tip

Connettore SWD

Il tipo e la posizione del connettore SWD varia sui diversi modelli. Su Pico e Pico W, è costituito da tre pin. Su Pico H e Pico WH, è una porta JST.

Programmare **Pico**

Collegheiamo Pico a Raspberry Pi 5 e scriviamo un programma MicroPython per far lampeggiare il LED integrato.

01 Per programmare Pico con MicroPython, useremo l'IDE Thonny. Con Raspberry Pi5 collegato a un monitor, apri Thonny – trovalo nel Menù (in alto a sinistra) > Programmazione.

02 Collega l'estremità più grande del cavo a una qualsiasi porta USB libera su Raspberry Pi 5. Tenendo premuto il pulsante BOOTSEL di Pico, collega l'altra estremità del cavo alla porta micro-USB di Pico per montarlo come unità disco.

03 Per installare il firmware MicroPython sul Pico, In Thonny, fai clic su "Python 3 Locale" in basso a destra della finestra. Seleziona "Installa MicroPython" dal menu a comparsa.

04 Nel menu successivo, fai clic sul menu a discesa "variante" e seleziona il modello di Pico per la versione corretta. Infine, fai clic sul pulsante "Installa" per eseguire il flashing del firmware MicroPython sul Pico.

05 Un messaggio nel riquadro Shell confermerà la versione di MicroPython installata sul Pico connesso. Ora sei pronto per iniziare a programmarlo.

06 Per utilizzare i pin GPIO di Pico, dobbiamo importare la classe Pin del modulo MicroPython `machine`. Aggiungi questo codice al riquadro principale di Thonny:

```
from machine import Pin
```

07 Poi, assegniamo una variabile (l'abbiamo chiamata `led`) al pin GPIO del LED su Pico e impostiamolo come uscita:

```
led = Pin("LED", Pin.OUT)
```

08 Per attivare/disattivare il LED, aggiungiamo:

```
led.toggle()
```

Fai clic sul bottone Esegui di Thonny per eseguire il programma e accendere il LED di Pico. Eseguilo di nuovo per disattivarlo.

09 Per salvare il tuo programma, fai clic sul bottone Salva di Thonny e nomina il file con il suffisso .py – il nostro lo abbiamo chiamato flick.py – e abbiamo deciso di salvarlo sul Raspberry Pi Pico.

10 Per alternare lo stato del LED a intervalli regolari, modifichiamo il codice. Sotto la riga 1, aggiungi quanto segue per importare la classe sleep dal modulo `time`:

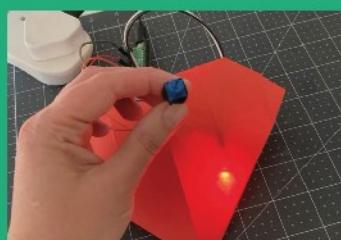
```
from time import sleep
```

11 Prima di `led.toggle`, aggiungi una nuova riga per creare un ciclo infinito:

```
while True:
```

Indenta (cioè rientra) `led.toggle` di quattro spazi (così sarà parte del ciclo).

12 Aggiungi un'altra riga indentata al ciclo per aggiungere un ritardo di un secondo:

```
sleep(1)
```

Ora esegui il codice e il LED dovrebbe accendersi e spegnersi ogni secondo.

I prossimi **Progetti Pico**

Ora che sai come programmare Pico, prova a realizzare questi progetti per principianti.

Beating Heart

Posiziona un LED all'interno di un cartoncino a cuore e usa un potenziometro per controllare la frequenza con cui lampeggia (la frequenza cardiaca).
magpi.cc/picoheart

Sensory Gadget

Crea un giocattolo interattivo come Picosaber o una candela digitale con più input e output di diversa tipologia.
magpi.cc/picogadget

Sound Machine

Collegando uno o più cicalini e controlli, Pico può riprodurre suoni o essere utilizzato come allarme.
magpi.cc/picosound